

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 08-09) – 7 Luglio 2009

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** della propria **HOME** directory che deve essere creato e avere nome **ESAME07Lug09-1-1**. FARE ATTENZIONE AL NOME DEL DIRETTORIO, in particolare alle maiuscole e ai trattini indicati. Verrà penalizzata l'assenza del direttorio con il nome indicato e/o l'assenza dei file nel direttorio specificato, al momento della copia automatica del direttorio e dei file. **ALLA SCADENZA DEL TEMPO A DISPOSIZIONE VERRÀ INFATTI ATTIVATA UNA PROCEDURA AUTOMATICA DI COPIA, PER OGNI STUDENTE DEL TURNO, DEI FILE CONTENUTI NEL DIRETTORIO SPECIFICATO.**
- 3) Il tempo a disposizione per la prova è di **120 MINUTI** per lo svolgimento di tutto il compito e di **75 minuti** per lo svolgimento della sola parte C.
- 4) Non è ammesso **nessun tipo di scambio di informazioni** né verbale né elettronico, pena la invalidazione della verifica.
- 5) L'assenza di commenti significativi verrà penalizzata.
- 6) **AL TERMINE DELLA PROVA È INDISPENSABILE CONSEGNARE IL TESTO DEL COMPITO (ANCHE IN CASO CHE UNO STUDENTE SI RITIRI): IN CASO CONTRARIO, NON POTRÀ ESSERE EFFETTUATA LA CORREZIONE DEL COMPITO MANCANDO IL TESTO DI RIFERIMENTO.**

Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** e una parte in **C**.

La parte in Shell deve prevedere due parametri: il primo deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system, mentre il secondo parametro deve essere un numero **K** *strettamente positivo e pari*. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono un numero **pari** di file (solo file e non direttori) *leggibili* maggiore o uguale a **K**. Si riporti sullo standard output sia il nome assoluto di tali direttori che il numero di file **F0... FN-1** che soddisfano la condizione. In ogni direttorio trovato, si deve invocare la parte in C, passando come parametri i file **F0... FN-1** trovati.

La parte in C accetta un numero variabile pari di parametri (si effettui il necessario controllo) che rappresentano nomi di file **F0...FN-1**. Il processo padre deve generare **N processi figli (P0 ... PN-1)**: ogni processo figlio è associato ad uno dei file **Fi**. I figli si devono considerare a coppie ordinate: ogni coppia è costituita dal processo di indice dispari Pd ($d = 1, 3, \dots N-1$) e dal processo di indice pari Pp ($p = 0, 2, \dots N-2$). La comunicazione in ognuna di tali coppie deve essere sempre dal processo dispari Pd al processo pari Pp. I processi della coppia devono eseguire **concorrentemente** leggendo i caratteri del loro file associato e contando quante occorrenze di **caratteri numerici** trovano. Al termine della lettura del proprio file associato, il processo dispari Pd comunica il valore numerico intero **Nn**, corrispondente al numero di occorrenze trovate, al processo pari Pp della sua coppia; il processo pari Pp deve confrontare il valore ricevuto dal processo dispari Pd con il proprio e quindi deve stampare su standard output chi dei due processi della coppia ha trovato il valore maggiore. Ogni processo figlio deve ritornare al padre il numero di occorrenze trovate.

Il padre, dopo che i figli sono terminati, deve stampare su standard output i PID di ogni figlio con il corrispondente valore ritornato.